Clustering of Sample Average Approximation for Stochastic Program

Lijian Chen[a]

[a]*Department of MIS, Operations Management, and Decision Sciences, University of Dayton, Dayton, OH 45469*

Abstract

We present an improvement to the Sample Average Approximation (SAA) method for two-stage stochastic program. Although the SAA has nice theoretical properties, such as convergence in probability and consistency, as long as the sample is large enough, the requirement on the sample size is always a concern for both academia and practitioners. Our clustering method employs the Maximum Volume Inscribed Ellipsoid (MVIE) to approximate the feasible set of each scenario and calculates a measure of similarity. The scenarios are clustered based on such a measure of similarity and our clustering method reduces the sample size considerably. Moreover, the clustering method will offer managerial implications by highlighting the mattering scenarios. The clustering method would be implemented in a distributed computational infrastructure with low-cost computers.

Keywords: Stochastic program, Sample Average Approximation, Maximum volume inscribed ellipsoid (MVIE), Clustering

1. Introduction

A standard formulation of the two-stage stochastic program is

$$\min_x \{c^T x + \mathbb{E}[Q(x,\xi)], x \in X\} \tag{1}$$

where

$$Q(x,\xi) := \inf_y \{q^T y : Wy \geq h - Tx, y \in Y\} \tag{2}$$

ξ represents random vectors and the expectation is taken with respect to the probability distribution of ξ. W is a deterministic matrix while h and T would be a function of ξ. Let $X \subset \mathbb{R}^n$ and when $Y \subset \mathbb{R}^m$, the problem is called stochastic linear program while $X \subset \mathbb{Z}^n$ or $Y \subset \mathbb{Z}^m$ indicates the problem is a stochastic *integer* program. This, in this paper, only stochastic linear program or stochastic integer program is concerned. One of the mostly adopted techniques to solve model (1) is named the Sample Average Approximation (SAA) by [6]. The main idea of SAA is to take N realizations, $\xi^1, \ldots, \xi^N$, of the random vector ξ to approximate the expected value function $\mathbb{E}[Q(x,\xi)]$ with its sample average $\frac{1}{N}\sum_{i=1}^N Q(x,\xi^i)$. We solve

$$\min_x \{c^T x + \frac{1}{N}\sum_{i=1}^N Q(x,\xi^i), x \in X\} \tag{3}$$

which is a deterministic problem and the computational cost is largely determined by the sample size N. Let $\hat{\nu}_N$ and $\hat{x}_N$ denote the optimal value and the optimal solution of problem (3), respectively; and ν^* and x^* represent the optimal value and the optimal solution of the true problem (1). $\hat{\nu}_N$ and $\hat{x}_N$ will converge to their counterparts as the sample size $N \to \infty$ regardless the distribution type of ξ, discrete or continuous, when the distribution has a finite support by [15], and when the distribution has an infinite support by [13, Chapter 5].

However, there is still one unanswered issue, the required sample size of SAA. The theoretical argument on determining the required sample size of the SAA is based on the set arguments and the Large Deviation theory. The theoretical argument is that, with high probability, the values of the sample average approximation and the true function are close to each other at a sufficiently dense set of points. The result is unsurprisingly impractical because the required sample size has been one of the primary impediments to SAA's implementation in practice. Even if some restrictive assumptions are imposed (see [14, Theorem 5.18]), it still requires a very large sample. In addition, the stochastic program has been recognized as #P-hard which indicates computationally intractability by [3]. The term #P-hard, rather than NP-hard, is used to describe a fact that the computer hardware will be overwhelmed by the number of scenarios required for the stochastic program. For stochastic program with integer recourse in particular, such a large sample size will lead to an unrealistically large deterministic integer program problem which is difficult to solve by most computer systems.

This argument leads to an immediate need of a smaller sample demanded by the SAA procedure. We propose a novel approach to reduce the sample size of the SAA. The idea of this paper is inspired by a fact that, within a pre-generated large, independent and identically distributed (iid) sample, there are many similar scenarios, i.e., excluding them from SAA procedure will lead to significant savings without considerably compromising the solution quality. Let us consider the following example:

Example 1. *From two raw materials, raw1 and raw2, we may simultaneously produce two different goods, prod1 and prod2. The unit costs of the raw materials are* $c = (c_1, c_2)' = (2, 3)'$. *The demands for the products are* $\xi = (\xi_1, \xi_2)'$ *and the production capacity is* $b = 100$. *If the values of* ξ_1 *and* ξ_2 *are deterministic, the production planning model is a plain integer program model.*

$$\min_{x_1,x_2} \{2x_1 + 3x_2 | x_1 + x_2 \leq 100,\ 2x_1 + 6x_2 \geq \xi_1,\ 3x_1 + 3x_2 \geq \xi_2,\ x_1 \geq 0, x_2 \geq 0\}$$

In principle, the clients expect the firm to satisfy demands. Very likely, according to the previously made production plan, the random components ξ_1 *and* ξ_2 *cause the event that the demands can not be covered by production. The amount of shortage has to be bought from the market and we assume that the costs per unit of lost sales products are* $q = (q_1, q_2)' = (7, 12)'$. *In the case of repeated execution, the best interest of the firm is to minimize the* expected *cost objective. This is the typical setting for the two-stage stochastic program by introducing the second-stage recourse variables,* $y_1(x_1, x_2, \xi_1)$ *and* $y_2(x_1, x_2, \xi_2)$. *We use the notation of* $y_1(x_1, x_2, \xi_1)$ *and* $y_2(x_1, x_2, \xi_2)$ *to emphasis the fact that the recourse decision depends on the first-stage variables and the realizations of uncertainty. We thus have the stochastic program model as follows:*

$$\begin{aligned}
&\min_{x_1,x_2}\ 2x_1 + 3x_2 + \mathbb{E}[Q(x_1, x_2, \xi_1, \xi_2)] \\
&\text{subject to: } x_1 + x_2 \leq 100, x_1 \geq 0, x_2 \geq 0, x_1 \text{ and } x_2 \text{ are integers} \\
&\text{where } Q(x_1, x_2, \xi_1, \xi_2) := \min\ 7y_1(x_1, x_2, \xi_1) + 12y_2(x_1, x_2, \xi_2) \\
&\text{subject to: } y_1(x_1, x_2, \xi_1) \geq \xi_1 - 2x_1 - 6x_2 \\
&\qquad y_2(x_1, x_2, \xi_2) \geq \xi_2 - 3x_1 - 3x_2 \\
&\qquad y_1(x_1, x_2, \xi_1) \geq 0, y_2(x_1, x_2, \xi_2) \geq 0. \\
&\qquad y_1(x_1, x_2, \xi_1), y_2(x_1, x_2, \xi_2) \text{ are integers.}
\end{aligned}$$

In order to adopt the SAA method, we generate a Monte-Carlo sample of size N, $\xi^1, \ldots, \xi^N$ *where* $\xi^i =$

$(\xi_1^i, \xi_2^i)'$. *The SAA solves*

$$\begin{aligned}
\min_{x_1,x_2}\ & 2x_1 + 3x_2 + \frac{1}{N}\sum_{i=1}^{N}[7y_1(x_1, x_2, \xi_1^i) + 12y_2(x_1, x_2, \xi_2^i)] \\
\textit{subject to: } & x_1 + x_2 \le 100 \\
& 2x_1 + 6x_2 + y_1(x_1, x_2, \xi_1^i) \ge \xi_1^i \\
& 3x_1 + 3x_2 + y_2(x_1, x_2, \xi_2^i) \ge \xi_2^i \\
& x_1 \ge 0, x_2 \ge 0, y_1(x_1, x_2, \xi_1^i) \ge 0, y_2(x_1, x_2, \xi_2^i) \ge 0 \\
& i = 1, \ldots, N \\
& x_1, x_2, y_1(x_1, x_2, \xi_1), y_2(x_1, x_2, \xi_2) \textit{ are integers.}
\end{aligned}$$

Let ξ_1, ξ_2 *be discrete uniform distributed random variables ranging* $310, 311, \ldots, 319$ *and* $292, 293, \ldots, 301$*, respectively. Consider the demands* ξ_1 *and* ξ_2 *are independent. The total number of scenarios is 100 with even probabilities and we solve the model with all the scenarios,*

$$x^* = [x_1^*; x_2^*]' = [70; 30]', \textit{ with the optimal value } 231.2.$$

The deterministic counterpart is an integer program with 202 *variables and* 201 *constraints. This model can be easily solved with additional information such as dual results and index of binding constraints. By selecting the scenarios associated with binding constraints, we identified the following scenarios which will yield an impact to the optimal solution. These scenarios are:*

Scenarios	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
ξ_1	310	311	312	313	314	315	316	317	318	319
ξ_2	300	300	300	300	300	300	300	300	300	300
Scenarios	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20
ξ_1	310	311	312	313	314	315	316	317	318	319
ξ_2	301	301	301	301	301	301	301	301	301	301

and we solve another deterministic counterpart with these 20 scenarios only for the optimal solution. The model is an integer program with 42 *variables and* 41 *constraints. The solution is*

$$x_{20}^* = [71; 29]', \textit{ with the optimal value } 232.$$

Now we adopt our clustering approach, which will be discussed in the later sections, to reduce the number of scenarios. We measure the similarity of these 100 scenarios and consolidate "similar" scenarios to obtain a much smaller sample. The scenarios in the newly obtained sample are not evenly likely. Based on these

Scenarios	#1	#2	#3	#4	#5	#6	#7
ξ_1	319	319	319	319	319	319	319
ξ_2	294	296	297	298	299	300	301
Probability	0.4	0.1	0.1	0.1	0.1	0.1	0.1

scenarios, we formulate an integer program with 16 *variables and* 15 *constraints. The optimal solution,* x_7^**, is*

$$x_7^* = [70; 30]', \textit{ with the optimal value } 231.2.$$

The obtained optimal solution is identical to the original and we would have sampled only 7 scenarios rather than 100 for this specific problem for the same optimal solution. The integer program would have been reduced from 202 integer variables with 201 integer constraints (without counting the non-negative constraints) to only 16 integer variables and 15 integer constraints. Given the fact that solving the integer program could be notoriously expensive, such a reduction on the model scale would be greatly appreciated by the practitioners.

We propose our clustering approach to reduce the required sample size of SAA without compromising the solution quality. The word "clustering" is to group a set of scenarios in such a way that the scenarios in the same group are more similar to each other than to these in other groups. The goal of this approach is to greatly simplify the calculations of the stochastic program, particularly nonlinear and or non-convex problems in which large sample size is a serious problem and an effective reduction of sample size makes a significant difference towards the improvement of computational saving. The managerial implication of this approach is to highlight a manageable number of scenarios which deserve more attention from the decision maker.

We would like to differ our approach from others which is also designed to reduce the number of scenarios, such as the scenario reduction method (see [2]), variance reduction methods (see [7]), and the Quasi-Monte-Carlo (see [9]) including Latin-Hyper Cube method (see [5]). Our approach is an improvement or a service patch to the SAA method. Despite the similar intentions to reduce the sample size, our approach only works on iid samples and thus our approach would inherit nearly all the nice theoretical properties of the SAA while most of other methods works on a non-iid sample ([9] uses iid sample as an exception). In fact, our approach is nowhere close to the methods such as scenario reduction, and variation reduction. In [10], we find a method that shares the similar goal with ours. In this paper, the authors are to find a bound on the non-smooth recourse function by strategies, such as the collinearity and the scenario selection. The scenario selection, in particular, is nearly the same as the scenario reduction method introduced earlier. We realize that our method and the method in [10] are considerably different because they are under different solution schemes: the sample average approximation and the stochastic decomposition.

Our approach is to attach a value to each sampled scenario as a measure of similarity and we will cluster similar scenarios to reduce the sample size. For each scenario, we construct a deterministic problem with polyhedral feasibility sets and approximate them by their Maximum Volume Inscribed Ellipsoid (MVIE). Then an orthogonal projection of this MVIE is used to obtained an ellipsoid that is free of the previous decisions. This approximate feasible region is used to solve the recourse problem, yielding the optimal values for each scenario in the pre-generated iid sample. The optimal value is then used as a measure of similarity. This measure of similarity meets two requirements: being a scalar, and being independent from previous decisions because the past decisions have been projected out.

We have good reasons to decline some seemingly good alternatives to fetch the measure of similarity. The first seemingly good alternative to our approach is to solve the single-scenario problem and use the optimal value as the measure of similarity. The solution includes both the first-stage variables and the recourse variables. Suppose two distinct scenarios i and j, the optimal solutions would be $[x_i^*; y_i^*]$ and $[x_j^*; y_j^*]$ respectively such that $x_i^* \neq x_j^*$, and the optimal values are ν_i^* and ν_j^*, respectively. If we use ν_i^* and ν_j^* as the measure of similarity which will be problematic because these values are determined by distinct first-stage decisions. Thus, this alternative to our approach fails to break the stage dependency and the optimal values would not be chosen as the measure of similarity of scenarios. The second seemingly good alternative is to adopt either Fourier-Motzkin or Gaussian eliminations to remove the dependency of previous decisions. The motive of this alternative is understandable that the elimination-based approaches only involve linear constraints and linear program which are known to be less expensive for decades. Nevertheless, the elimination-based approaches have complexity issues. When a polyhedron becomes increasingly complicated, the number of vertexes may explode and leads to a non-polynomial complexity.

Our approach is justified by both theoretical and computational results. For distinct scenarios, the polyhedral feasible sets are expected to be geometrically distinct as well. John's theorem in [4] shows that there exists a *unique* ellipsoid for each feasible set. Thus, we can establish a mathematical bijection from the scenarios to their MVIEs. The calculation of the measure of similarity is neither expensive nor time-consuming because of the advance of the semi-definite program. There are multiple matured and still actively updated software packages, such as SeDuMi (see [11]) and SDPA (see [16]). These implementations are all open-source and perform well on a large variety of platforms.

Let us do a brief cost-benefit analysis for our clustering approach. The cost of our approach is that we need to calculate the MVIE for each sampled scenario and the solution technique is convex optimization. The clustering procedure can be completed on multiple, low-cost, average desktops deployed *in parallel* rather

than an expensive supercomputer. According to our numerical study in Section 5, the calculation of the measure of similarity would be completed within a timely manner on an average desktop. The benefit of our approach is to reduce the numbers of variables and constraints to their fractions. In practice, our approach is more promising that practitioners can start the clustering Monte-Carlo sample at an early time to "select" scenarios as many as possible. Once a solution is demanded, the selected scenarios can be instantly feed into the SAA to obtain solution with an error bound estimate. Moreover, the clustering method will highlight a set of mattering scenarios which deserve more attention in the decision process.

The reminder of this paper is organized as follows. We present the unique representation of a scenario by its Maximum Volume Inscribed Ellipsoid (MVIE) in Section 2. In Section 3, we show the calculation of the measure of similarity of scenario. In Sections 4, we show that the clustering approach preserves nice theoretical properties of the SAA for stochastic program with integer recourse in particular. We show our numerical results in Section 5 and we conclude our approach with remarks in Section 6.

2. Quantifying scenarios by ellipsoids

In this section, we need to quantify each scenario by an ellipsoidal object. Prior to the discussion, we need to clarify the notational settings of the stochastic program with integer recourse. When the feasible set X in (1) is a bounded polyhedron, i.e., $X := \{x|Ax \leq b, x \geq 0\}$ where A has only finite numbers of rows and columns and the uncertainty is modeled through $K < \infty$ possible scenarios at probability $p_k, k = 1, \ldots, K$, the two-stage stochastic program with recourse is presented as:

$$\begin{aligned} &\min_x \; c'x + \sum_{k=1}^{K} p_k Q(x, \xi^k) \\ &\text{subject to} \;\; Ax \leq b, x \geq 0 \end{aligned} \tag{4}$$

where $Ax \leq b,\ x \geq 0, x \in \mathbb{R}^n$ or $x \in \mathbb{Z}^n$ are the resource capacity constraints and

$$\begin{aligned} Q(x, \xi^k) = \; &\min_{y_k} \; q_k' y_k \\ &\text{subject to} \;\; Wy_k \leq h_k - T_k x, y_k \geq 0. \end{aligned} \tag{5}$$

The matrices $T_k, h_k,\ y \in \mathbb{R}^m$ or $y \in \mathbb{Z}^m$, and $q_k \geq 0$ are functions of ξ and W is a fixed matrix. In the literature, this problem is called two-stage stochastic program with fixed recourse and $Q(x, \xi)$ is called the recourse function. The distribution of ξ can be either discrete or continuous and there would be finitely many or infinitely many realizations. We adopt the scenario generation method to approximate the original distribution with a finite number of realization $K < \infty$, $\xi^1, \ldots, \xi^K$ and we can call each realization a scenario. We have:

$$\begin{aligned} \min_{x, y_1, \ldots, y_K} \;\; &c'x + \sum_{k=1}^{K} p_k q_k' y_k \\ \text{subject to} \;\; &Ax \leq b, x \geq 0 \\ &Wy_k \leq h_k - T_k x, y_k \geq 0, k = 1, \ldots, K \end{aligned}$$

where p_k is the probability of scenario ξ^k. The optimal solution is x^* with the optimal value ν^*. In order to adopt the SAA method to obtain a meaningful result, we need the following assumptions:

Assumption 1. *When the first-stage variables are discrete, the set of first-stage decisions is non-empty and finite. When the first-stage variables are continuous, the set of first-stage decisions is non-empty, compact, and polyhedral.*

Assumption 2. *When the first-stage variables are continuous, $Q(x, \xi^k)$ is finite, $k = 1, \ldots, K$. When the first-stage variables are discrete, the recourse function $Q(x, \xi)$ is measurable and $\mathbb{E}|Q(x, \xi)|$, $\mathbb{E}(Q^2(x, \xi))$ are finite for every x.*

Assumption 3. *For any first-stage x, the feasible set of recourse variables is non-empty and finite.*

These assumptions imply that the expected value and variance of the objective function are finite for all feasible x. The SAA method solves this problem in two steps. First, a iid sample, $\xi^1, \ldots, \xi^N, N >> K$, is generated. Second, the SAA problem

$$\begin{aligned} \min_{x, y_1, \ldots, y_N} \quad & c'x + \frac{1}{N}\sum_{i=1}^{N} q_i' y_i \\ \text{subject to} \quad & Ax \leq b, x \geq 0, x \in X \\ & Wy_i \leq h_i - T_i x, y_i \geq 0, i = 1, \ldots, N \end{aligned} \tag{6}$$

is solved and its optimal first-stage solution is x_N^* with the optimal value ν_N^* when $\xi^1, \ldots, \xi^N$ are treated equally likely.

When we sample *only one* scenario ξ^i with $p_i = 1$ and we have:

$$\begin{aligned} \min_{x_i, y_i} \quad & c'x_i + q_i' y_i \\ \text{subject to} \quad & Ax_i \leq b, x_i \geq 0, x_i \in X \\ & Wy_i \leq h_i - T_i x_i, y_i \geq 0 \end{aligned}$$

where x_i and y_i are the first-stage and recourse variables, respectively. We define

Definition 1.

$$P_i([x_i; y_i]) := \{[x_i; y_i] | Wy_i \leq h_i - T_i x_i, y_i \geq 0, y_i \in \mathbb{R}^m\}$$

is named the $(n+m)$-dimension scenario polyhedron *of the i^{th} scenario.*

$P_i([x_i; y_i]), i = 1, \ldots, N$ are polyhedra with respect to $[x_i; y_i], i = 1, \ldots, N$ and we use P_i as its short form. We need to remark that $P_i([x_i; y_i])$ is for both the stochastic linear program and the stochastic integer program. For the program with integer variables, in order to calculate the measure of similarity, we need to relax the discrete feasible set into a polyhedron. Moreover, we need to assume:

Assumption 4. *$P_i([x_i; y_i])$ needs to be full-dimensional and bounded.*

We employ the ellipsoidal objects to differentiate scenarios $P_i, P_j, i \neq j$ because of the following definitions:

Definition 2. *An ellipsoid E in $\mathbb{R}^{n+m}$ is an affine image of the unit ball $B_{n+m} = \{u \in \mathbb{R}^{n+m} : ||u|| \leq 1\}$, that is,*

$$E = \{c + Su : u \in \mathbb{R}^{n+m}, ||u|| \leq 1\}, \text{ or } E = \{x \in \mathbb{R}^{n+m} : ||S^{-1}(x - c)|| \leq 1\}$$

where $S \in \mathbb{R}^{(n+m)\times(n+m)}$ is a symmetric, non-singular, and positive definite matrix.

This assumption 4 is rather important. If $P_i([x_i; y_i])$ is not full-dimensional, the maximum volume inscribed ellipsoid of this polyhedron will be degenerate and have a volume of 0. The matrix of the ellipsoid may not be positive-definite. For most two-stage stochastic programs with recourse, the second stages are usually bounded. When $P_i([x_i; y_i])$ is not bounded, there will be no maximum volume inscribed ellipsoid associated with it. Thus, we have to artificially impose a bound. In such a case, the artificial bounds should meet two requirements: first, although the bound is imposed on a un-bounded polyhedron, this bound should not affect the optimal solution with the given objective; second, the same bound should be applied to all the scenarios. Later, we will discuss through examples on the purpose of these two requirements which ensure that the bounds only play a minimum role for the calculation of the measure of similarity.

Now we present the well-known John's theorem:

Theorem 1. *Let C be a convex, bounded, and nonempty polyhedron in $\mathbb{R}^{n+m}$. There exists an ellipsoid of maximum volume inscribed in C.*

We can find the proof of this theorem in many articles and books, e.g. [1] and we omit the proof.

Theorem 2 (John's theorem)**.** *The maximum volume ellipsoid inscribed in C is unique and there is an ellipsoid with the same center but scaled by a factor $n+m$ contains C.*

The proof of the above theorem is in many articles, e.g. [4] and we omit the proof.

For any pair of distinct scenarios ξ^i and ξ^j, their scenario polyhedra P_i and P_j are geometrically distinct to each other. By applying Theorems 1 and 2, the MVIEs of P_i and P_j are distinct with each other and the MVIEs, centered at $c(i)$ and $c(j)$ with symmetric positive-definite matrix $S(i)$ and $S(j)$, are denoted by $(S(i), c(i))$ and $(S(j), c(j))$. Likewise, we use $(\bar{S}(i), c(i))$ to denote the ellipsoid scaled by a factor $n+m$ from $(S(i), c(i))$ that contains P_i. We have

$$(S(i), c(i)) \subset P_i \subset (\bar{S}(i), c(i))$$

The MVIE, $(S(i), c(i))$, is calculated by solving a Semi-Definite Program (SDP). Suppose we can represent the i^{th} scenario polyhedron by ℓ linear inequality of variables x_i and y_i, i.e., $\{a_j'[x_i; y_i] \leq b_j, j = 1, \ldots, \ell\}$, the SDP problem thus is

$$\begin{aligned} &\min_{S(i), c(i)} \ \log(det(S(i)^{-1})) \\ &\text{subject to: } ||S(i)a_j||_2 + a_j' c(i) \leq b_j, j = 1, \ldots, \ell \end{aligned} \tag{7}$$

The construction of the above model is from the geometric property of the non-degenerate ellipsoid. The volume of an ellipsoid is proportional to the value of the determinant of $S(i)$. Thus, the objective is to minimize the logarithmic function of the determinant of $S^{-1}(i)$. The constraints are imposed to ensure the MVIE will be inscribed to a polyhedron $\{a_j'[x_i; y_i] \leq b_j, j = 1, \ldots, \ell\}$. Thus, this model is a convex optimization and the overall complexity is $O[(n+m)^3]$. This model per se, can be efficiently solved, thanks to the fast advance of interior-point methods since the 1990s. These implementations are all open-source and perform well on a large variety of platforms.

3. A measure of similarity for distinct scenario polyhedra

We now construct a measure of similarity among distinct scenario polyhedra. It does not make sense to compare $P_i([x_i; y_i])$ and $P_j([x_j; y_j])$ when $x_i \neq x_j$ because the solution of stochastic program is a decision for all K scenarios. Thus, we need to set the first-stage decision to be $x_0 \in \mathbb{R}^n$ for all the scenarios. x_0 is rather symbolic and it will be projected out. P_i degenerates to a m-dimensional polyhedron with respect to y_i,

$$P_i([x_0; y_i]) := \{[x_0; y_i] | W y_i \leq h_i - T_i x_0, y_i \geq 0, y_i \in \mathbb{R}^m\}$$

at a realized random vector $\xi^i, i = 1, \ldots, N$. We define

$$\begin{aligned} \tilde{Q}(x_0, \xi^i) := &\min_{y_i} \ q_i' y_i \\ &\text{subject to} \ \ W y_i \leq h_i - T_i x_0, y_i \geq 0 \end{aligned} \tag{8}$$

as the value of Polyhedral Feasible Region Recourse (PFRR) of ξ^i. The center $c(i)$ can be rewritten as $[c_x(i); c_y(i)]$ where $c_x(i) \in \mathbb{R}^n$, $c_y(i) \in \mathbb{R}^m$. $c_x(i)$ is a n-dimensional coordinate and $c_y(i)$ is a m-dimensional coordinate. $c_x(i)$ and $c_y(i)$ represent the projections of $c(i)$ onto X and Y, respectively. We define:

$$\begin{aligned} \eta_i(x_0) := &\min_{y_i} \ q_i' y_i \\ &\text{subject to} \ \ \begin{pmatrix} x_0 - c_x(i) \\ y_i - c_y(i) \end{pmatrix}' S^{-2}(i) \begin{pmatrix} x_0 - c_x(i) \\ y_i - c_y(i) \end{pmatrix} \leq 1 \end{aligned} \tag{9}$$

the Ellipsoidal Feasible Region Recourse (EFRR). The feasible set of EFRR is $(S(i), c(i))$ with $x_i = x_0$ imposed. The value of $\eta_i(x_0)$ is uniquely determined by the value of x_0 and $S^{-2}(i)$. For a given x_0, we can compare $\eta_i(x_0)$ and $\eta_j(x_0)$ for the similarity between distinct scenarios ξ^i and ξ^j. Nevertheless, $\eta_i(x_0)$ is still a function of x_0 and the value of $\eta_i(x_0)$ should not be used as the measure of similarity when solving a stochastic program. An ideal measure of similarity should be a scalar which is independent of the first-stage decision.

If we take the *orthogonal projection* of P_i onto the affine space of y_i, we may completely remove the first-stage variables and formulate a smaller but non-empty feasible set of y_i. Likewise, the orthogonal projection of the $(n+m)$-dimensional ellipsoid $(S(i), c(i))$ onto Y, denoted as $S_y(i) \in \mathbb{R}^m$, is another ellipsoid of y_i only. When we approximate the feasible set by $S_y(i)$, we obtain a first-stage-variable-free part of the recourse function. We use this value, denoted by η_i, as the measure of similarity of scenarios. η_i is the optimal value of the following model:

$$\begin{aligned} \eta_i = \min_{y_i} \ & q_i' y_i \\ \text{subject to} \ & y_i \in S_y(i). \end{aligned}$$

The calculation of the orthogonal projection of $S_y(i)$ is quite straightforward. Consider the scenario i and its MVIE $(S(i), c(i))$ where $S(i)$ is the matrix and $c(i)$ is the center of MVIE. The $(S(i), c(i))$ is

$$\{c(i) + S(i)u | u \in \mathbb{R}^{n+m}, ||u|| \leq 1\}$$

The affine space is given by $\{Mt\}$ where $M = [0; I]$ is orthogonal such that $0 \in \mathbb{R}^{n \times m}$ and I is the $m \times m$ identity matrix and t is a vector of m parameters. The orthogonal projection of a general point $[x_i; y_i]$ of MVIE onto the affine space is $MM'[x_i; y_i]$. We thus obtain

$$S_y(i) = \{MM'(c(i) + S(i)u) | ||u|| \leq 1\}.$$

Now let the *singular value decomposition* (SVD) of $M'S(i)$ be

$$M'S(i) = U[0; \Sigma]V'.$$

where Σ is the diagonal matrix with diagonal elements Σ_{ii} such that $\dfrac{1}{\Sigma_{ii}}$ is the length of the i^{th} principal semi-axis of the MVIE. Both U and V are $(n+m) \times (n+m)$ orthogonal matrices. Then

$$M'S(i)u = U\Sigma\tilde{w}$$

where $\tilde{w}$ denotes the corresponding m elements of $w := V'u$. Note $||u|| \leq 1$ implies $||\tilde{w}|| \leq 1$. Thus we have

$$S_y(i) = \{M(M'c(i) + U\Sigma\tilde{w}) | \ ||\tilde{w}|| \leq 1\}. \qquad \text{(The orthogonal projection)}$$

The above equation, $S_y(i)$, is for an m-dimensional ellipsoid onto the affine space of y_i. The $S_y(i)$ which is calculated by the following steps:

Step 1. Calculate $S_{temp} = (S(i)^{-1})'$.
Step 2. Calculate $S_{temp2} = MM'S_{temp}$ to obtain a $m \times m$ matrix.
Step 3. Calculate the inverse, $S_y(i) = (S_{temp2})^{-1}$.

Thus, we have the orthogonal projection of the MVIE, $(S_y(i), c_y(i))$. We exclude the first-stage variables from the model and the feasible region becomes $(S_y(i), c_y(i))$,

$$\min_{y_i} \{q_i' y_i \mid ||S_y(i)^{-1}(y_i - c_y(i))|| \leq 1\} \tag{10}$$

with optimal value η_i. In Step 1, the inverse of a matrix will be stable because of the Assumption 4. Since each scenario polyhedron and its MVIE are full-dimensional, the matrix $S(i)$ will be non-singular.

The value η_i is an ideal measure of similarity of scenarios. First, η_i is uniquely associated with the i^{th} scenario because of John's Theorem. Second, η_i is no longer a function of x_0. Third, *any* MVIE projection at a specific x_0 will be a subset of the orthogonal projection, i.e.,

$$\left\{ y_i \middle| \begin{pmatrix} x_0 - c_x(i) \\ y_i - c_y(i) \end{pmatrix}' S^{-2}(i) \begin{pmatrix} x_0 - c_x(i) \\ y_i - c_y(i) \end{pmatrix} \leq 1 \right\} \subset S_y(i)$$

and we thus have

$$\eta_i \leq \eta_i(x_0)$$

for any feasible x_0. Likewise, any MVIE projection at x_0 will be a subset of the polyhedron of y_i at x_0, i.e.

$$\left\{ y_i \middle| \begin{pmatrix} x_0 - c_x(i) \\ y_i - c_y(i) \end{pmatrix}' S^{-2}(i) \begin{pmatrix} x_0 - c_x(i) \\ y_i - c_y(i) \end{pmatrix} \leq 1 \right\} \subset \left\{ y_i \middle| Wy_i \leq h(\xi^i) - Tx_0, y_i \geq 0 \right\}$$

which implies

$$\tilde{Q}(x_0, \xi^i) \leq \eta_i(x_0)$$

By the establishment of η_i, we define:

Definition 3. $\tilde{\epsilon}_i(x_0) := \eta_i - \tilde{Q}(x_0, \xi^i)$.

We use the following example to demonstrate the calculation of $\tilde{Q}(x_0, \xi^i), \eta_i(x_0)$, and η_i.

Example 2. *Consider a single-scenario problem in which the first-stage variable is $x \in \mathbb{R}$ and so is the second-stage variable $y \in \mathbb{R}$. The problem is*

$$\begin{aligned} &\min\ 0x + \tilde{Q} \\ &\textit{subject to: } 0 \leq x \leq 1 \end{aligned}$$

and

$$\begin{aligned} \tilde{Q} = &\min\ y \\ &\textit{subject to: } |y| \leq x \end{aligned}$$

There is only one scenario and the scenario polyhedron is

$$\{[x; y] | x \leq 1,\ -x \leq 0,\ -x + y \leq 0,\ -x - y \leq 0\}$$

and the MVIE is centered at $[\frac{2}{3}; 0]$ *with matrix*

$$\begin{pmatrix} \frac{1}{3} & 0 \\ 0 & \frac{\sqrt{3}}{3} \end{pmatrix}$$

The projection at x_0 is an ellipsoid of y, i.e., a segment. For example, the projection of MVIE at $x_0 = \frac{2}{3}$ will be the segment $\{[\frac{2}{3}; y]| - \frac{\sqrt{3}}{3} \leq y \leq \frac{\sqrt{3}}{3},\ x_0 = \frac{2}{3}\}$. When either $x_0 = \frac{1}{3}$ or $x_0 = 1$, the projections at x_0 will be two points: $\{[x_0, y] = [\frac{1}{3},\ 0]\}$ and $\{[x_0, y] = [1,\ 0]\}$, respectively. The orthogonal projection of the MVIE onto the affine space of y will be the

$$\{y| - \frac{\sqrt{3}}{3} \leq y \leq \frac{\sqrt{3}}{3}\}$$

by solving Model (10). *In this example, we know that*

$$\eta_1(\frac{1}{3}) = \eta_1(1) = 0,\ \eta_1(\frac{2}{3}) = -\frac{\sqrt{3}}{3},\ \eta_1 = (-\frac{\sqrt{3}}{3})$$

$$\tilde{Q}(\frac{1}{3}, \xi^1) = -\frac{1}{3},\ \tilde{Q}(\frac{2}{3}, \xi^1) = -\frac{2}{3},\ \tilde{Q}(1, \xi^1) = -1$$

We organize the value of $\tilde{\epsilon}_i(x_0)$ *at different values of* x_0.

x_0	$\tilde{Q}(x_0, \xi^1)$	η_1	$\tilde{\epsilon}_1(x_0)$
0	0	$-\sqrt{3}/3$	$-\sqrt{3}/3$
1/3	$-1/3$	$-\sqrt{3}/3$	$(1-\sqrt{3})/3$
2/3	$-2/3$	$-\sqrt{3}/3$	$(2-\sqrt{3})/3$
1	-1	$-\sqrt{3}/3$	$(3-\sqrt{3})/3$

In this example, the projection of MVIE at a given x_0 *has two major problems. First, there are some feasible* x_0 *which is outside of the MVIE and* $\eta_i(x_0)$ *does not exist. For example, when* $x_0 \in [0, \frac{1}{3})$, $\eta_1(x_0)$ *does not exist. Second, the error between* $\tilde{Q}(x_0, \xi^i)$ *and* $\eta_i(x_0)$ *could be considerably large. For example, when* $x_0 = 1$, *we have* $\eta_1(1) = 0$ *but* $\tilde{Q}(1, \xi^1) = -1$ *that the error is maximized. In comparison to the projection of MVIE at a given* x_0, *the orthogonal projection of MVIE has neither of the problems above and we have a unique value of* $\eta_1 = -\frac{\sqrt{3}}{3}$ *as the measure of similarity of this scenario polyhedron.*

4. Consolidation of similar scenarios

In this section, we show our approach to cluster similar scenarios for stochastic linear program and stochastic integer program. Consider the SAA problem with a sample of size N, $\xi^1, \ldots, \xi^N$,

$$\begin{aligned} \min_{x, y_1, \ldots, y_N} \quad & c'x + \frac{1}{N}\sum_{k=1}^{N}[\eta_k - \tilde{\epsilon}_k(x)] \\ \text{subject to: } & Ax \le b,\ x \ge 0 \end{aligned} \tag{11}$$

with the optimal solution x_N^* and the optimal value $\nu_N^* := \nu_N(x_N^*)$. There are two "similar" scenarios $i, j, i \neq j$ such that $|\eta_i - \eta_j| < \delta$ for a $\delta > 0$. We cluster the scenario j with the scenario i. This consolidation is to change the sample $\xi^1, \ldots, \xi^N$ to $\xi^1, \ldots, \xi^{j-1}, \xi^i, \xi^{j+1}, \ldots, \xi^N$ and we have

$$\begin{aligned} \min_{x, y_1, \ldots, y_N} \quad & c'x + \frac{1}{N}\sum_{k \neq i,j}[\eta_k - \tilde{\epsilon}_k(x)] + \frac{2}{N}[\eta_i - \tilde{\epsilon}_i(x)] \\ \text{subject to: } & Ax \le b,\ x \ge 0 \end{aligned} \tag{12}$$

The difference between objectives of (11) and (12) at a given x is bounded by:

$$\frac{1}{N}\left|\eta_i - \eta_j + \tilde{\epsilon}_j(x) - \tilde{\epsilon}_i(x)\right| < \frac{\delta}{N} + \frac{1}{N}\left|\tilde{\epsilon}_j(x) - \tilde{\epsilon}_i(x)\right|$$

For a given $\delta > 0$, we can cluster a certain number of scenarios. Suppose we cluster $[N^{1-\kappa}], 0 < \kappa < 1$ (the notation $[\ \cdot\ ]$ represents the operation of round up to the closest integer) times and there is $N - [N^{1-\kappa}]$ scenarios left in the model. Let $\mathcal{J}$ represents the set of indices of scenarios being clustered and $\mathcal{I}$ represents the set of indices of scenarios remaining in the model.

Definition 4. *The scenario of* $\mathcal{I}$ *are named representative scenarios.*

Assumption 5. *For every scenario of $\mathcal{J}$ or a separately generated scenario from certain simulation procedure, we can always find a representative scenario of $\mathcal{I}$ such that*

$$|\eta_i - \eta_j| < \delta, i \in \mathcal{I}, j \in \mathcal{J}$$

at a given $\delta > 0$.

The optimal solution is $\tilde{x}_N^*$ and the optimal value becomes $\tilde{\nu}_N^* := \tilde{\nu}_N(\tilde{x}_N^*)$ and $\tilde{\nu}_N$ represents the objective function of the clustered model. The consolidation is denoted as $(i,j) \in \mathcal{I} \times \mathcal{J}$. The difference from ν_N^* is bounded by:

$$|\tilde{\nu}_N(x) - \nu_N(x)| < \frac{[N^{1-\kappa}]}{N}\delta + \frac{1}{N}\left|\sum_{(i,j)\in\mathcal{I}\times\mathcal{J}}[\tilde{\epsilon}_i(x) - \tilde{\epsilon}_j(x)]\right| < \delta + \frac{1}{N}\left|\sum_{(i,j)\in\mathcal{I}\times\mathcal{J}}[\tilde{\epsilon}_i(x) - \tilde{\epsilon}_j(x)]\right|.$$

The goal of our clustering approach is to cluster similar scenarios from a pre-generated Monte Carlo sample and preserve nice theoretical properties, such as consistency and exponentially fast convergence rate of the SAA. Thus, our approach is rather an improvement to the SAA. The value of $|\tilde{\nu}_N(x) - \nu_N(x)|$ is the measure of solution quality in comparison to the SAA with a sample size of N. In order to show the preservation of nice theoretical properties of SAA, we first need to prove that the both $|\sum_{(i,j)\in\mathcal{I}\times\mathcal{J}}(\eta_i - \eta_j)|$ and $|\sum_{(i,j)\in\mathcal{I}\times\mathcal{J}}(\tilde{\epsilon}_i(x) - \tilde{\epsilon}_j(x))|$ are bounded. Since P_i is bounded, we have

Proposition 1. *There exists $\mathbb{D}$ such that $|\tilde{\epsilon}_i(x) - \tilde{\epsilon}_j(x)| < \mathbb{D}$ for any x.*

Proposition 2. *For a sample of size N, we cluster up to $[N^{1-\kappa}]$ times, we have*

$$|\tilde{\nu}_N(x) - \nu_N(x)| < \frac{1}{[N^\kappa] - 1}\delta + \frac{1}{[N^\kappa] - 1}\mathbb{D}$$

for $1 > \kappa > 0$ and $\forall x$ such that $Ax \leq b, x \geq 0$.

Proof:

$$\begin{aligned}
|\tilde{\nu}_N(x) - \nu_N(x)| &= \frac{1}{N}\left|\sum_{(i,j)\in\mathcal{I}\times\mathcal{J}}[\eta_i - \eta_j + \tilde{\epsilon}_j(x) - \tilde{\epsilon}_i(x)]\right| \\
&\leq \frac{1}{N}\sum_{(i,j)\in\mathcal{I}\times\mathcal{J}}\left|\eta_i - \eta_j\right| + \frac{1}{N}\sum_{(i,j)\in\mathcal{I}\times\mathcal{J}}\left|\tilde{\epsilon}_j(x) - \tilde{\epsilon}_i(x)\right| \\
&\leq \frac{[N^{1-\kappa}]}{N}\delta + \frac{[N^{1-\kappa}]}{N}\mathbb{D} = \frac{1}{[N^\kappa] - 1}\delta + \frac{1}{[N^\kappa] - 1}\mathbb{D}
\end{aligned}$$

□

Theorem 3. *For an iid sample of size N, we cluster up to $[N^{1-\kappa}]$ times,*

$$|\tilde{\nu}_N^* - \nu_N^*| \to 0$$

as $N \to \infty$ with probability 1.

Proof: It is clear that both $\tilde{\nu}_N(x)$ and $\nu_N(x)$ are convex with respect to x. By definition, we have

$$\tilde{\nu}_N(\tilde{x}_N^*) \leq \tilde{\nu}_N(x_N^*) \text{ and } \nu_N(x_N^*) \leq \nu_N(\tilde{x}_N^*)$$

By Proposition 2, when N is large enough and any $\epsilon > 0$, we have

$$\tilde{\nu}_N(\tilde{x}_N^*) \leq \tilde{\nu}_N(x_N^*) \leq \nu_N(x_N^*) + \epsilon \text{ and } \nu_N(x_N^*) \leq \nu_N(\tilde{x}_N^*) \leq \tilde{\nu}_N(\tilde{x}_N^*) + \epsilon$$

Thus, when N is large enough, $|\tilde{\nu}_N^* - \nu_N^*| \leq \epsilon$ with probability 1. □

There is another appearance of the above result that for a sample of size N, we cluster up to $[N^{1-\kappa}]$ times, we have

$$\lim_{N\to\infty} \mathbb{P}(|\tilde{\nu}_N^* - \nu_N^*| > \epsilon) = 0$$

where $1 > \kappa > 0, \epsilon > 0$.

We now present the impact of our clustering approach on the rate of convergence of the SAA. On the other hand, our approach does reduce the original sample size N to its fraction $N - N^{1-\kappa}, 1 > \kappa > 0$. It is fair to say that by adopting our approach, we are now capable of obtaining an optimal solution comparable to x_N^* with a significantly small sample of $N - N^{1-\kappa}$.

Let $\mathcal{S}_N^\epsilon$ and $\mathcal{S}^\epsilon$ of ϵ-optimal solutions of the SAA and true problems, respectively. Let $\mathcal{S}$ represent the set of optimal solutions of the true problem. Both $\mathcal{S}_N^\epsilon$ and $\mathcal{S}^\epsilon$ are nonempty and finite for any $\epsilon > 0$. When pursuing different accuracy on the SAA and the true problem with accuracy $\gamma > 0$ and $\epsilon > 0$, respectively such that $\gamma \leq \epsilon$, the event $\{\mathcal{S}_N^\gamma \subset \mathcal{S}^\epsilon\}$ means that the solution of the SAA provides an ϵ-optimal solution of the true problem. We need the following definition

Definition 5. *Let $X := \{x|Ax \leq b, x \geq 0\}$. $u(x)$ is a mapping from $X\backslash\mathcal{S}^\epsilon$ into the set $\mathcal{S}$ that $u(x) \in \mathcal{S}$ for all $x \in X\backslash\mathcal{S}^\epsilon$ such that for $\epsilon^* := \min_{x\in X\backslash\mathcal{S}^\epsilon} \nu(x) - \nu^*, \epsilon^* \geq \epsilon$,*

$$\nu(u(x)) \leq \nu(x) - \epsilon^* \text{ for all } x \in X\backslash\mathcal{S}^\epsilon$$

along with the following assumption:

Assumption 6. *For every $x \in X\backslash\mathcal{S}^\epsilon$ the moment generating function of the random variable $Y(x,\xi) := c'u(x) + \tilde{Q}(u(x),\xi) - c'x - \tilde{Q}(x,\xi)$ is finite valued in a neighborhood of $t = 0$.*

We thus have

Theorem 4. *Let ϵ and γ be non-negative numbers such that $\gamma \leq \epsilon$. Then,*

$$1 - \mathbb{P}(\mathcal{S}_N^\gamma \subset \mathcal{S}^\epsilon) \leq Ke^{-N\eta(\gamma,\epsilon)} \tag{13}$$

where

$$\eta(\gamma,\epsilon) := \min_{x\in X\backslash\mathcal{S}^\epsilon} \mathcal{I}_x(-\gamma) \tag{14}$$

where $\mathcal{I}_x(\cdot)$ denote the rate function of the random variable $Y(x,\xi)$. With the assumption 6, $\eta(\gamma,\epsilon) > 0$ and K is the number of scenarios of the true problem.

Proof: The proof is in [12, page 373].

The new sample will no longer be iid and the following argument has nothing to do with the Large Deviation theory but to apply Theorem 4. Let $\beta := \frac{1}{[N^\kappa]-1}\delta + \frac{1}{[N^\kappa]-1}\mathbb{D}$ be the error bound estimate of $[N^{1-\kappa}]$ consolidations and we solve Model (21) for τ-optimal solutions. We denote $\mathcal{S}_{N,\kappa}^{\tau,\beta}$ the set of resulting optimal solution and we would calibrate the parameters γ, and κ of the clustering to satisfy:

$$\mathcal{S}_{N,\kappa}^{\tau,\beta} \subset \mathcal{S}_N^\gamma. \tag{15}$$

Therefore, we have

Theorem 5. *For a Monte-Carlo sample of size N, we cluster $N^{1-\kappa}$ times to control the error bound at $\beta = \frac{1}{[N^\kappa]-1}\delta + \frac{1}{[N^\kappa]-1}\mathbb{D}$ and solve Model (21) with an accuracy τ. If (15) is satisfied, then*

$$\limsup_{N\to\infty} \frac{1}{N} \log\left[1 - \mathbb{P}(\mathcal{S}_{N,\kappa}^{\tau,\beta} \subset \mathcal{S}^\epsilon)\right] \leq -\eta(\gamma,\epsilon) \tag{16}$$

Proof: Since $\mathcal{S}_{N,\kappa}^{\tau,\beta} \subset \mathcal{S}_N^{\gamma}$, we have

$$\mathbb{P}(\mathcal{S}_{N,\kappa}^{\tau,\beta} \subset \mathcal{S}^{\epsilon}) \geq \mathbb{P}(\mathcal{S}_N^{\gamma} \subset \mathcal{S}^{\epsilon}) \tag{17}$$

and by Theorem 4,

$$1 - \mathbb{P}(\mathcal{S}_{N,\kappa}^{\tau,\beta} \subset \mathcal{S}^{\epsilon}) \leq Ke^{-N\eta(\gamma,\epsilon)}. \tag{18}$$

An immediate consequence of (18) is (16) and we are done. □

Since the stochastic linear program can be efficiently solved by many other techniques, such as stochastic decomposition and the plain sample average approximation, we need to show the merit of our clustering method when solving stochastic integer program. Now, we consider $Y \subset \mathbb{Z}^m$.

$$\begin{aligned} Q(x_0, \xi^i) = \min_{y_i} \ & q_i' y_i \\ \text{subject to} \ & Wy_i \leq h_i - T_i x_0, y_i \geq 0. \\ & y_i \text{ are integers.} \end{aligned} \tag{19}$$

$Q(x_0, \xi^i)$ is the optimal value of an integer program while $\tilde{Q}(x_0, \xi^i)$ is the optimal value of a linear program with integer constraints relaxed. Along the direction of q_i, feasible sets $S_y(i)$, P_i, and $\{y_i \in \mathbb{Z}^m | Wy_i \leq h - Tx_0\}$ have the optimal values η_i, $\tilde{Q}(x_0, \xi^i)$, and $Q(x_0, \xi^i)$, respectively. We define

Definition 6. $\epsilon_i(x_0) := \eta_i - Q(x_0, \xi^i)$.

Because all the feasible sets are bounded or finite (it may be very large number of points), we can show that there exists $\mathbb{D}$ such that

$$|\epsilon_i(x_0) - \epsilon_j(x_0)| < \mathbb{D}, \text{ for any pair within the } N \text{ scenarios.}$$

By using similar argument as above, we can show that our clustering approach clusters similar scenarios with respect to the measure of similarity η_i and η_j without violating the consistency of the SAA, i.e.

$$\lim_{N \to \infty} \mathbb{P}(|\tilde{\nu}_N^* - \nu_N^*| > \epsilon) = 0 \tag{20}$$

for any $\epsilon > 0$. The argument will be very similar and we omit it.

We need to remark that we still use the

$$\frac{1}{N} \left| \sum_{(i,j) \in \mathcal{I} \times \mathcal{J}} [\eta_i - \eta_j + \tilde{\epsilon}_j(\tilde{x}_N^*) - \tilde{\epsilon}_i(\tilde{x}_N^*)] \right|$$

as the error bound estimate rather than updating $\tilde{\epsilon}_i(\tilde{x}_N^*)$ and $\tilde{\epsilon}_j(\tilde{x}_N^*)$ with $\epsilon_i(\tilde{x}_N^*)$ and $\epsilon_j(\tilde{x}_N^*)$ for two reasons. First, calculating the value $\epsilon_i(\tilde{x}_N^*)$ and $\epsilon_j(\tilde{x}_N^*)$ will be expensive. In fact, the only model that we solved with integer variables is the SAA model with most scenarios clustered. Second, the gap between $Q(x_0, \xi^i)$ and $\tilde{Q}(x_0, \xi^i)$ is not considerably large indeed.

We now present the clustering approach the following 8 steps:

Step 1. Generate a large enough Monte-Carlo sample, i.e., $\xi^1, \ldots, \xi^N$. By relaxing the integer constraints, we thus have N convex and bounded scenario polyhedra, $P_1, \ldots, P_N$.

Step 2. We calculate $(S(1), c(1)), \ldots, (S(N), c(N))$ and $(S_y(1), c_y(1)), \ldots, (S_y(N), c_y(N))$.

Step 3. Calculate the value of $\eta_i, i = 1, \ldots, N$ and sort them in an ascending order. Determine the value of $\delta > 0$.

Step 4. We have a pool of η_i with a finite support. We evenly partition this support into $B(\delta)$ disjoint segments of length δ (known as bins). We position each η_i into a bin by its value. Suppose there are $n, \ldots, n_{B(\delta)}$ scenarios in these bins, respectively, such that $\sum_{i=1}^{B(\delta)} n_i = N$.

Step 5. For the $j^{th}, j = 1, \ldots, B(\delta)$ bin, we choose the scenario with the bin's median value (if n_j is even, then choose $(n_j/2+1)^{th}$ scenario) and place it into the set $\mathcal{I}$. Otherwise, the scenario will be placed into the set $\mathcal{J}$.

Step 6. Calculate $p_j := n_j/N, j = 1, \ldots, B(\delta)$. Solve the following model

$$\begin{aligned} \min \;\; & c'x + \sum_{i \in \mathcal{I}} p_i q_i' y_i \\ \text{subject to: } & Ax \leq b, \; x \geq 0 \\ & Wy_i \leq h_i - T_i x, y_i \geq 0, i \in \mathcal{I} \end{aligned} \tag{21}$$

to obtain the solution $\tilde{x}_N^*$.

Step 7. Extract the error bound estimate $\frac{1}{N}\left|\sum_{(i,j)\in\mathcal{I}\times\mathcal{J}}[\eta_i - \eta_j + \tilde{\epsilon}_j(\tilde{x}_N^*) - \tilde{\epsilon}_i(\tilde{x}_N^*)]\right|$ to assess the solution quality. This value represents the error *bound* estimate between $\nu_N(x)$ and $\tilde{\nu}_N(x)$ at the point $\tilde{x}_N^*$.

Step 8. Record all scenario consolidations.

In practice, Model (21) performs very well in terms of solution quality. In Step 5, we choose the scenario with the bin's median value to control the value of $\frac{1}{N}\sum_{(i,j)\in\mathcal{I}\times\mathcal{J}}[\eta_i - \eta_j]$ by taking the advantage of cancelations among η_i values. Similarly, the cancelation among $\epsilon_i(x)$ will lead to, very likely, a much lower value of $\sum_{(i,j)\in\mathcal{I}\times\mathcal{J}}[\epsilon_i(x) - \epsilon_j(x)]$ than its bound $[N^{1-\kappa}]\mathbb{D}$. Thus, the number of consolidations would usually be greater than $[N^{1-\kappa}]$ because our analysis assumes that there is no cancellations among $\epsilon_i(x), i = 1, \ldots, N$. This is for the theoretical analysis and a more likely case is that the cancellations will play an important role to lower the value of $\frac{1}{N}\left|\sum_{(i,j)\in\mathcal{I}\times\mathcal{J}}[\eta_i - \eta_j + \tilde{\epsilon}_j(\tilde{x}_N^*) - \tilde{\epsilon}_i(\tilde{x}_N^*)]\right|$. Thus, we would be able to cluster more scenarios.

We present a cost-benefit analysis for adopting our approach for the stochastic program with integer recourse. For an iid sample of size N, the cost of our approach is to solve N semi-definite program to calculate the MVIE $(S(i), c(i))$ and its orthogonal projection $(S_y(i), c_y(i))$ for $i = 1, \ldots, N$. The cost seems formidable and expensive but the reality suggests otherwise. Thanks to the advance of convex optimization, the cost is less of a concern because both semi-definite program and orthogonal projection can be efficiently solved within a timely manner. In addition, the clustering method is able to identify the mattering scenario rather than treating all the scenarios equally. In addition to the solution, the decision makers may interested in identifying several scenarios which deserves more attention.

The benefit of our approach is that we greatly reduce the number of scenarios without compromising the solution quality. A reduction of scenario implies the reduction of the number of integer variables and constraints. For example, in example 1, our approach reduces the original integer program of 202 integer variables and 201 integer constraints to another instance of 16 integer variables and 15 integer constraints. We have more numerical results in Section 5 to illustrate the benefit of our approach. Consider the fact that the integer program will always be NP-hard, the benefit of reducing the scale of an integer program to its fraction can be easily justified. In the real-world implementation, practitioners can start the clustering Monte-Carlo sample at an early time to "select" scenarios. Once a solution is demanded, the selected scenarios can be instantly input into the Model (21) to obtain solution with an error bound estimate.

5. Numerical results

We first present a detailed illustration of the clustering procedure to Example 1 with $K = 100$ evenly likely scenarios. We organize this example by the following steps.

Step 1. Solve the following model to calculate MVIE. For example, the 100^{th} scenario is $\xi_1 = 319, \xi_2 = 301$.

$$\min \;\; \log det(S^{-1}) \text{ subject to:} ||Sa_i|| + a_i' c \leq b_i, i = 1, 2, \ldots, 9$$

where $A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ -2 & -6 & -1 & 0 \\ -3 & -3 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ and the right-hand-side is $\begin{pmatrix} 100 \\ -319 \\ -301 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 2 \end{pmatrix}$ We impose the last two constraints, $y_1 \leq 2, y_2 \leq 2$, are required to bound the polyhedron. These constraints will not void the optimal solution in any sense. The MVIE for this scenario is

$$c(100) = \begin{pmatrix} 35.1668 \\ 64.7223 \\ 1.0000 \\ 1.6664 \end{pmatrix}, \; S(100) = \begin{pmatrix} 24.9024 & -24.8309 & 0.0024 & -0.0230 \\ -24.8309 & 24.9023 & -0.0024 & -0.0228 \\ 0.0024 & -0.0024 & 1.0000 & -0.0000 \\ -0.0230 & -0.0228 & -0.0000 & 0.3321 \end{pmatrix}$$

Step 2. We calculate the orthogonal projection for all the scenarios. For example, the 100^{th} scenario's orthogonal projection is

$$c_y(100) = \begin{pmatrix} 1.0000 \\ 1.6664 \end{pmatrix}, \; S_y(100) = \begin{pmatrix} 1.0000 & -0.0000 \\ -0.0000 & 0.3174 \end{pmatrix}$$

Step 3. Calculate $\eta_k, k = 1, \ldots, 100$ by Model (10). For example, $\bar{\nu}_{100}$ is calculated by solving

$$\begin{aligned} &\min \; [7; 12]' y \\ &\text{subject to: } || \begin{pmatrix} 1.0000 & -0.0000 \\ -0.0000 & 0.3174 \end{pmatrix}^{-1} (y - [1; 1.6664]) || \leq 1 \end{aligned}$$

We have $\bar{\nu}_{100} = 19.02795$.

Step 4. We $\delta = 0.05$ and we have 7 bins. The selected scenarios are

Bins	probability	η_k	index	$\tilde{\epsilon}_k(\tilde{x}^*_N)$	ξ_1	ξ_2
1	0.4	13.17462419	93	13.174624	319	294
2	0.1	13.22648319	45	13.226483	319	296
3	0.1	13.27733964	46	13.277340	319	297
4	0.1	13.37219897	47	13.372199	319	298
5	0.1	13.60167657	48	13.601677	319	299
6	0.1	16.64961449	49	16.649614	319	300
7	0.1	19.02760473	50	7.027605	319	301

Table 1: Selected scenarios for Example 1. The column of $\epsilon_k(\tilde{x}^*_N)$ is calculated in Step 6.

Step 5. Solve Model (21) to obtain $\tilde{\nu}^*_N$ and optimal solution $\tilde{x}^*_N$. In this example, we have

$$\tilde{x}^*_{100} = [70.25; 29.75], \tilde{\nu}^*_{100} = 230.95$$

Step 6. Calculate $\tilde{\epsilon}_k(\tilde{x}^*_N) = \eta_k - \nu_k(\tilde{x}^*_N)$ for all the scenarios. For example $\tilde{\epsilon}_{100}(\tilde{x}^*_N) = 7.027594$.

Step 7. Calculate the error term $\frac{1}{N} \left| \sum_{(i,j) \in I \times \mathcal{J}} [\eta_i - \eta_j + \epsilon_i(\tilde{x}^*_N) - \epsilon_j(\tilde{x}^*_N)] \right|$. In this example, the error term is -0.1842.

Step 8. Record all scenario consolidations.

From the analysis, we present the fact that our clustering greatly reduces the number of similar scenarios. In this example, the obtained optimal coincides with the true optimal and we reduce the number of scenarios from 100 to 7 at a *possible* cost of -0.1842 which counts 0.075% of the optimal value. Moreover, the clustering approach provides a better model understanding that only 7 scenarios count most. Thus, these selected scenarios deserve more attention from the decision maker.

We now present results of another well-known stochastic program, "LandS". The problem description is in [8] and the dimensional information is summarized in Table 2. The recourse variables are imposed to be integers. We implement the clustering approach on the platform of Debian Linux with Matlab R2013b and

Name	Application	Scenarios	First-Stage variables	Recourse variables
LandS	Electricity Planning	1×10^6	$x \in \mathbb{R}^4$	$y \in \mathbb{Z}^{12}$

Table 2: Test Problem Dimensions, x denotes the first-stage variables and y denotes the recourse integer variables

SeDuMi package as the SDP solver on an average computer with 4 Gigabyte memory. The clustering approach is applied to a Monte-Carlo sample of size $N = 20,000$ by which a qualified optimal solution is obtainable. We organize the results in Table 3 in columns of problem, δ, the number of bins, $\frac{1}{N}|\sum_{(i,j)\in I\times\mathcal{J}} \eta_i - \eta_j|$, $\frac{1}{N}|\sum_{(i,j)\in I\times\mathcal{J}} \tilde{\epsilon}_i(\tilde{x}^*_N) - \tilde{\epsilon}_j(\tilde{x}^*_N)|$, and $\tilde{\nu}^*_N$. If $\kappa = 0.05$, there will be $20000^{0.95} = 12189$ consolidations. In

Problem	δ	# of bins	$\frac{1}{N}\lvert\sum_{(i,j)\in I\times\mathcal{J}} \eta_i - \eta_j\rvert$	$\frac{1}{N}\lvert\sum_{(i,j)\in I\times\mathcal{J}} \tilde{\epsilon}_j(\tilde{x}^*_N) - \tilde{\epsilon}_i(\tilde{x}^*_N)\rvert$	$\tilde{\nu}^*_N$
LandS	1	152	0.002289	12.24	237.11
LandS	0.5	288	0.000326	9.29	239.49
LandS	0.05	1841	0.0000985	2.07	238.64

Table 3: The numerical results for the problem "Lands" with $N = 20,000$.

practice, we can try a greater number of consolidations. For example, when we choose $\delta = 0.05$ and cluster 18159 scenarios, the possible error will be as much as 2.07, or 0.86% of the optimal value.

We now show the gain of our approach. Without clustering, the plain SAA will use an iid sample of size $N = 20,000$. The equivalent integer program will have 240,004 variables and 240,000 variables among them are integers. There will be 140,002 constraints. If we adopt the clustering approach with $\delta = 0.05$ and we cluster 12,189 scenarios, the resulting integer program will have 93,736 variables and 54,679 constraints at a cost of *up to* 0.86% of the optimal value. If we cluster 18,159 scenarios, the equivalent integer program will have 22,096 variables and 12,889 constraints. The cost of our clustering is minor. On an average computer, the computational time of solving SDP, calculating the orthogonal projection, and calculating the measure of similarity combined will be ranging from 10 to 12 seconds. For $N = 20,000$ scenarios, it will take up to 240,000 seconds (less than 67 hours). Since the clustering can be deployed to computers in parallel. Suppose we have 10 average computers for clustering, it will only take less than 7 hours. Given the well-known difficulty of integer program, such a reduction on the problem scale would always be well justified. In addition, the clustering time may be further shortened because the prototype of our approach is implemented on Matlab which is known to be slower than packages coded in efficient languages such as C++.

6. Conclusion

In this paper, we propose the clustering method for the stochastic linear program and stochastic integer program. The key idea is to attach a value to each sampled scenario as the measure of similarity and the sampled scenarios with similar values would be clustered as one representative scenario to reduce the sample size. We show that the clustering approach inherits nice theoretical properties of the SAA. The clustering approach will lead to a significantly small but representative sample to deliver timely solution without

compromising the solution quality. The implementation of clustering can be a distributed computational infrastructure in which the clustering Monte-Carlo sample is completed by low-cost computers deployed in parallel rather than expensive supercomputers. The benefit of clustering is to reduce the scale of stochastic program to its fraction. In our illustrative examples, the nearly 90% of the integer variables and constrains would be clustered. In comparison to its benefit, the cost of clustering is rather minor, thanks to the advance of convex optimization since the 1990s. The clustering method also highlights a subset of the scenarios which need more attention from the decision maker because these scenarios will generate significant impacts to the optimal solution than the rest of scenarios.

[1] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge Univ Pr, 2004.

[2] J. Dupačová, N. Gröwe-Kuska, and W. Römisch, *Scenario reduction in stochastic programming*, Mathematical programming, 95 (2003), pp. 493–511.

[3] M. Dyer and L. Stougie, *Computational complexity of stochastic programming problems*, Mathematical Programming. A Publication of the Mathematical Programming Society, 106 (2006), pp. 423–432.

[4] O. Güler and F. Gürtuna, *The extremal volume ellipsoids of convex bodies, their symmetry properties, and their determination in some special cases*, Arxiv preprint arXiv:0709.0707, (2007).

[5] T. Homem-de-Mello, *On rates of convergence for stochastic optimization problems under non-independent and identically distributed sampling*, SIAM Journal on Optimization, 19 (2008), pp. 524–551.

[6] A. J. Kleywegt, A. Shapiro, and T. Homem-de-Mello, *The sample average approximation method for stochastic discrete optimization*, SIAM Journal on Optimization, 12 (2001), pp. 479–502 (electronic).

[7] M. Koivu, *Variance reduction in sample approximations of stochastic programs*, Mathematical programming, 103 (2005), pp. 463–485.

[8] J. Linderoth, A. Shapiro, and S. Wright, *The empirical behavior of sampling methods for stochastic programming*, Annals of Operations Research, 142 (2006), pp. 215–241.

[9] P. LEcuyer and C. Lemieux, *Recent advances in randomized quasi-monte carlo methods*, in Modeling uncertainty, Springer, 2005, pp. 419–474.

[10] W. Oliveira, C. Sagastizábal, and S. Scheimberg, *Inexact bundle methods for two-stage stochastic programming*, SIAM Journal on Optimization, 21 (2011), pp. 517–544.

[11] D. Peaucelle, D. Henrion, Y. Labit, and K. Taitz, *Users guide for sedumi interface 1.04*, LAAS-CNRS, Toulouse, (2002).

[12] A. Ruszczynski and A. Shapiro, eds., *Stochastic Programming*, vol. 10 of Handbooks in Operations Research and Management Science, Elsevier, 2003.

[13] T. Santoso, S. Ahmed, M. Goetschalckx, and A. Shapiro, *A stochastic programming approach for supply chain network design under uncertainty*, 2003. Published: Stochastic Programming E-Print Series, http://www.speps.org.

[14] A. Shapiro, D. Dentcheva, and A. Ruszczyski, *Lectures on stochastic programming: modeling and theory*, vol. 9, Society for Industrial and Applied Mathematics, 2009.

[15] A. Shapiro and T. Homem-de-Mello, *On the rate of convergence of optimal solutions of monte carlo approximations of stochastic programs*, SIAM Journal on Optimization, 11 (2000), pp. 70–86 (electronic).

[16] M. Yamashita, K. Fujisawa, and M. Kojima, *Implementation and evaluation of sdpa 6.0 (semidefinite programming algorithm 6.0)*, Optimization Methods and Software, 18 (2003), pp. 491–505.